Detecting Edges with Bi-Directional Tracing Across Multiple Scales

Ting Huang
School of Computer Science and Technology
Soochow University
Suzhou, China
20165227046@stu.suda.edu.cn

Baojiang Zhong
School of Computer Science and Technology
Soochow University
Suzhou, China
bjzhong@suda.edu.cn

***Abstract*—For our human vision system, natural images are understood via a progressive and hierarchical process; that is, a multi-scale analysis. Motivated by this fact, a multi-scale algorithm for edge detection is proposed in this paper. The novelty of the proposed algorithm lies in that a bi-directional tracing strategy is used, which first conducts a *forward tracing* (FT) operation in coarse-to-fine direction and then a *backward confirm* (BC) operation in fine-to-coarse direction across multiple scales. With the FT operation, the location accuracy of detected edges is improved; with the BC operation, spurious edges are removed. The edge map generated with this bi-directional tracing strategy can therefore have a better quality than that generated by the existing multi-scale edge detectors. Extensive simulation results have shown that our new algorithm is superior to a number of state-of-the-art edge detection algorithms in both subjective visual assessment and objective measurement.**

***Index Terms*—Edge detection; Multiple scales; Gaussian smoothing; Edge tracing; Edge confirm; Bi-directional tracing.**

I. Introduction

Edges are highly descriptive features of an image, and edge detection plays an important role in various computer vision tasks. An edge refers to a set of pixels that change most prominently in the image. Detected edges can be widely used in feature extraction [1], image recognition [2], image segmentation [3] and so on. Over the past decades, many algorithms for edge detection have been proposed, including the classic operators [4], [5], the optimal algorithms [6], [7], the multi-scale methods [8], the adaptive smoothing based methods [9], the fuzzy mathematic-based algorithms [10] and the neural-network-based algorithms [11], [12].

The performance of an edge detector usually depends on the adjustment of a *scale* parameter. Unfortunately, it is quite difficult to automatically determine an optimal scale for edge detection [13], [14]. If a high scale is used, principal contours that have long length or strong contrast can be perceived, while short-length and week-contrast edges will be ignored, resulting in a high precision but a low recall in detection performance. If a low scale is used, all edges might be detected; however, spurious edges could also be generated, resulting in a high recall but a low precision.

This research is supported by the National Natural Science Foundation of China under grants 61572341.

For *human visual system* (HVS), edge detection is a multi-scale process. The HVS has the ability to extract useful information from different scales and yield an integrated result of edge detection. Motivated by this, some algorithms are proposed in the literature to detect edges at varying scales or via a multi-scale scheme [15]–[18]. In the present work, a multi-scale edge detection algorithm incorporated with a *bi-directional tracing* strategy is proposed and investigated. First, edge maps are generated by edge detectors at several given scales. Second, a *forward tracing* (FT) operation (in the coarse-to-fine direction across multiple scales) is conducted: for each edge presented at the high scale, its neighboring low-scale counterparts are traced and recorded by using a set of anchors. With the FT operation, the localization accuracy of detected edges is improved. Third, a *backward confirm* (BC) operation (in the fine-to-coarse direction across multiple scales) is conducted: each edge obtained after the previous FT operation is confirmed by a backward tracing, i.e., to trace it from the low scales to the high scales. If an edge does not have a counterpart in a high scale, it will be discarded as a spurious edge. Compared with other multi-scale algorithms (e.g., [15], [16]) and the existing single-scale methods, our proposed algorithm can deliver superior performance.

In Section II, the existing multi-scale edge detection methods are briefly reviewed. In Section III, the new algorithm is presented and discussed. In Section IV, extensive simulation experiments are conducted, with which the proposed algorithm and a number of state-of-the-art algorithms are evaluated and compared subjectively and objectively. Finally, in Section V the paper is concluded.

II. Related Work

The current multi-scale edge detection methods, as well as their multi-scale mechanism, are introduced in the following.

It is observed that single-scale algorithms are difficult to balance between accurate edge location and robustness to noise. To address this dilemma, various multi-scale methods have been proposed. The multi-scale concept in image processing is achieved by an increasing level of smoothing of the input image [19]. In a multi-scale edge detection, the scale parameter is in general taken to be the standard deviation of a Gaussian kernel, when the input image is smoothed.

978-1-7281-1312-8/18/$31.00 ©2018 IEEE

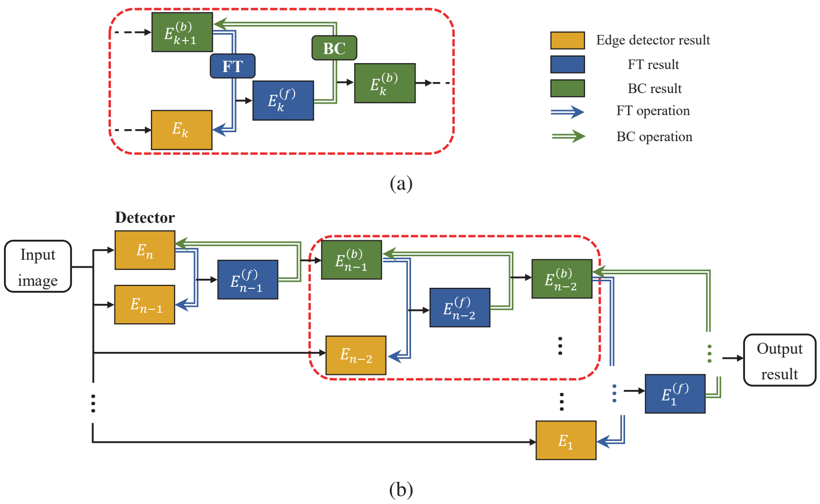


Fig. 1. Framework of the proposed algorithm. (a) A basic bi-directional tracing step conducted on two adjacent scales, which is comprised of a forward tracing (FT) operation (being shown as the blue double solid line), and a backward confirm (BC) operation (being shown as the green double solid line); (b) The edge detection process by using the bi-directional tracing step in a recursive manner.

One typical kind of multi-scale edge detection is to detect edges at multiple given scale individually and then take the averaged results as the final output [16], [20]. With the edge maps obtained at different scales, the final edge map can also be produced by a multi-scale fusing. So various methods fusing multi-scale information need to be explored. For that, a scale multiplication which multiplies the responses at the two scales is used in [15]. A method based on the RCF network constructed an image pyramid of the input image to conduct multi-scale method [2].

Another kind of multi-scale edge detection is to select a local scale for each pixel or each subregion [21], [22].

Edge tracking across multiple scales is also mentioned in many literatures [8], [12], [23]. Liu et al. employed a coarse-to-fine supervision strategy to reduce false positives [12]. Lopez-Molina et al. also introduced coarse-to-fine edge tracking in the Gaussian scale-space, specifically, edges could be tracked in the form of scale-descending edge pixel matching [8]. Sumengen et al. used a fine-to-coarse strategy which favors edges exist at multiple scales [23].

Learning-based edge detection methods (such as the structured edge (SE) detector [28]) represent an interesting trend in this research direction. However, according to our research experience, they are more suitable to be treated as shape contour detectors, and they have close relationship with image segmentation methods. The edges produced by using a learning-based method could be helpful in solving some tasks. However, in some other tasks a traditional edge detection could be more helpful instead.

Different from the methods introduced above, bi-directional edge tracing is proposed to fuse edge information. In our strategy, edges are tracked down to edge anchors. Anchors in ED [24] correspond to peaks of the gradient map, ours are a set of seed points which equally distribute on all edges.

In this work, our multi-scale method focuses on the first kind which is the most studied so far. We perform edge detector (e.g. Canny) at several given scales to get the edge map of the corresponding scale, and then combine the information across multiple scales.

III. The Proposed Algorithm

A. Problems of Single-Scaled Edge Detection

Gaussian smoothing has been used as an effective approach to suppress unwanted information such as noise in edge detection [7]. Denote the Gaussian filter as

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{1}$$

where σ is its standard deviation and can be treated as a scale parameter. Let $f(x, y)$ be the input image, its smoothed version takes the form

$$f_z(x, y) = G_\sigma(x, y) * f(x, y) \tag{2}$$

where "$*$" stands for the convolution operation. Unfortunately, small value of scale σ causes the smoothing effect not effective and those unwanted information still exists. On the other hand, if the scale value is set to be large, useful information will be suppressed, besides those unwanted information.

B. The Proposed Multi-scale Algorithm

Different to the traditional multi-scale strategy, in this paper we propose a novel multi-scale method, called the *bi-directional tracing*. In detail, let E_k $(k = 1, 2, \ldots)$ be the edge map generated at the k-th scale, a *forward tracing* (FT) operation is defined as

$$E_k^{(f)} = F(E_k, E_{k+1}) \tag{3}$$

and a *backward confirm* (BC) operation is defined as

$$E_k^{(b)} = B\left(E_k^{(f)}, E_{k+1}\right) \tag{4}$$

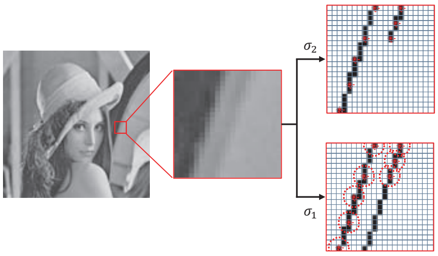
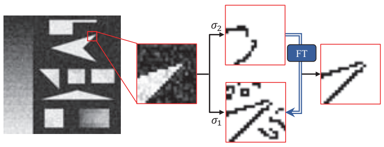
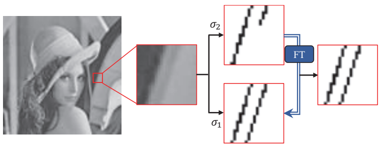


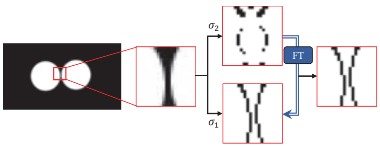
Fig. 2. How to conduct the FT operation in detail, where $\sigma_1 = 1$, $\sigma_2 = 1.5$ in Canny. Red asterisks denote the chosen anchors, and red dotted circles indicate a searching region with a radius of 2 pixels for tracing the edge around the anchors.



(a)



(b)



(c)

Fig. 3. Effects of the FT operation, where σ_1 is the low scale, σ_2 is the high scale. (a) FT for recovering the sharp corner; (b) FT for finding the vanished edges; (c) FT for revising the fused edges.

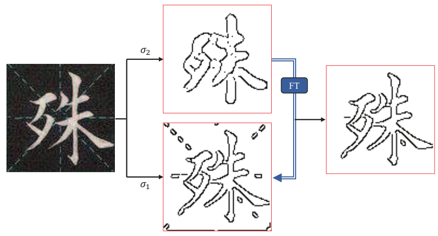
where $E_k^{(f)}$ represents the edge map yielded after the FT operation from the $(k+1)$-th scale to the k-th scale, and $E_k^{(b)}$ represents the edge map yielded after the BC operation from the FT to the $(k+1)$-th scale.

In Fig. 1(a), a basic bi-directional tracing step conducted on two scales is illustrated, which is comprised of a FT operation and a BC operation. Fig. 1(b) shows the process of our proposed multi-scale algorithm by implementing the bi-directional tracing step recursively across multiple scales.

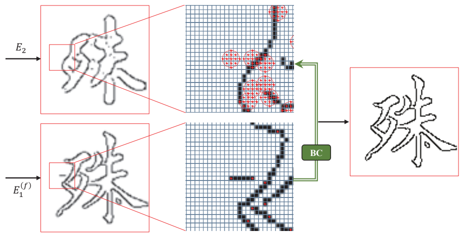
C. The Bi-Directional Tracing Strategy

In the following, it will be described how to conduct the bi-directional tracing and how the edge detection quality can be improved.

Fig. 2 shows the FT operation in detail. Perform the edge detector at given scales for one image, in Fig. 2, one part of 'Lena' image gets two Canny edge maps, first select equidistant anchors on every edge at high scale, then locate the positions at low scale as the same coordinate as anchors, and search the edge pixels which have distance less than 2 pixels to the positions. If the edge at low scale contains the searched edge pixels and is traced by an anchor, return it to produce FT edge map.



(a)

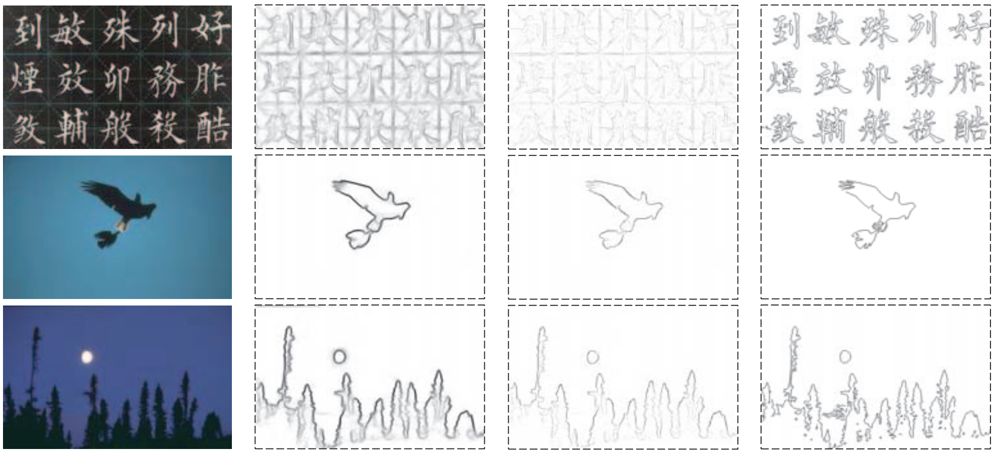


(b)

Fig. 4. How to conduct the BC operation. (a) An example to show that spurious edges still remain after FT, where $\sigma_1 = 1.5$, $\sigma_2 = 4$; (b) Result of conducting the BC operation, $E_1^{(f)}$ represents the FT edge map, E_2 is the high scale edge map.

The effects of the FT operation are revealed in Fig. 3. In Fig. 3(a), the input image contains sharp corners and one of them has been highlighted. To recover the sharpness of the corner at σ_2, a set of equidistant points are selected as anchors from the corresponding edge. We trace the anchors from σ_2 to σ_1 by using a searching ratio equal to 2 pixels. It is seen that the yield edge map contains a much shaper corner than that observed used high scale parameter. In Fig. 3(b), the highlighted image portion contains two adjacent edges. At a high scale σ_2 one of the edge is incomplete due to its low contrast (after Gaussian smoothing its contrast becomes even lower). By tracing them to σ_1, the edge shows complete length and the edge map therefore achieves higher accuracy. Two edges in Fig. 3(c) are fused because high scale σ_2, by FT operation, trace them to the corresponding edges at low scale σ_1 to revise the fused edges.

During the above FT operation, while the accuracy of edges can be improved, spurious edges could also be yielded due to noise effect. An example of an ancient calligraphy is illustrated in Fig. 4(a). At a high scale the detected edges show rounded corners and the background of the image causes some spurious edges which are of noise effect. However, at a low scale more strong spurious edges appear as Gaussian smoothing. After the FT operation, most spurious edges observed at the low scale have been removed (as shown in Fig. 4(a)), however, the FT result edge map still remains spurious edges because edges at different scale will shrinkage or collapse. To solve this problem, a *backward confirm* (BC) operation is performed as shown in Fig. 4(b). We trace every edge with anchors in the edge map yielded after the FT operation back to the high scale for confirm. If two associated edges (one is from σ_1 and the



(a) (b) (c) (d)

Fig. 5. Visual comparison. (a) Original image, first row is an ancient calligraphy, the second and third ones are two images taken from BSDS500 [27]; (b) SE [28] result; (c) SE result after non-maximal suppression; (d) The proposed algorithm.

other is from σ_2) have little difference, then the one presented in σ_2 is confirmed as a true edge, which will be passed to the next step for further processing or finally, be taken as an output edge. Otherwise, if two related edges shows great difference, then both of them are treated as spurious edges and will be completely removed from the edge candidate set.

With the FT and the BC operation, a bi-directional tracing step is performed. One can see from Fig. 4 that the final edge map has much improved quality compared with those yielded at each of the two scales alone. In detail, the detect edges show more shaper corners and no spurious edges are detected.

D. Main Steps of The Proposed Algorithm

The main steps of the proposed algorithm are described in the following Algorithm 1. The total number of scales is n, and therefore $n-1$ bi-directional tracing steps will be recursively implemented.

IV. Experimental Results and Performance Evaluations

In this section, the experimental results are presented and the performance of the proposed algorithm is compared with that of a number of state-of-the-art edge detection algorithms [7], [24]–[26] in both noise-free and noisy cases.

A. Visual Comparisons

We first test our algorithm and compare it with the typical learning-based edge detector proposed in [28] by using three test images (as shown in Fig. 5(a)). Fig. 5(b) shows the results yielded by [28], with the input images. Note that these edge maps are edge probability maps. Even with a non-maximal suppression, the edges in these maps are hard to be traced continuously (the edge tracing operation is usually needed in a system for the follow-up steps), as shown in Fig. 5(c). This is a limitation of the current leaning-based detectors. Fig. 5(d) shows the performance of our proposed detector. While such binary edges are easy to be traced, one can see that they are more precise than those presented in Fig. 5(b) or Fig. 5(c).

Algorithm 1 Multi-Scale Edge Detection with Bi-Directional Tracing

Input: An image and n given scales $\sigma = \{\sigma_1, ..., \sigma_n\}$.

Output: The edge map of the input image.

Step 1. Perform the edge detector (e.g. Canny) at n given scales individually;

Step 2. Denote the edge set yielded at the scale σ_i $(i = 1, ..., n)$ as $c_i = \left\{c_i^{(1)}, ..., c_i^{(m)}\right\}$ and let $l_i^{(j)}$ be the length of $c_i^{(j)}$, $(j = 1, ..., m)$. We select s anchors that equally distributed on the edge, where s is equal to the root length of the edge;

Step 3. Perform the FT operation to improve the edge accuracy. For the k-th anchor at the j-th edge (at the scale σ_i, $j = 1, ..., m$, $k = 1, ..., s$), we search the edge pixels (at the scale σ_{i-1}) which have distance less than 2 to this anchor. Return all the edges containing the searched pixels, which are treated as candidates of the low-scale counterpart of the edge;

Step 4. Perform the BC operation for edge confirming. For the j-th edge of the edge map obtained after Step 3, we also choose a set of equally-distant anchors. For the k-th anchor, record the counterpart of the anchor (at the scale σ_i), and then search the edge pixels q_i which have less than 2 to the counterpart of anchor. If q_i does not exist, denote the j-th edge be the spurious edge and discard it.

B. Objective Performance Evaluations

In the following we evaluate the performance of our algorithm with the existing edge detectors objectively. The objective evaluation metrics include the precision P, the recall R, the F-measure and the Pratt's Figure of Merit (FOM).

The F-measure is computed by

$$F_\alpha = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R} \quad (5)$$

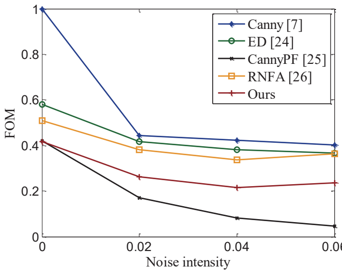


Fig. 6. Performance of algorithms under three levels of noise.

where α determines which one of P and R is more important. In our comparison, we use the value of $F_{0.5}$ which means P and R have the same importance.

The FOM metric is computed by

$$FOM = \frac{1}{\max\left(N_i, N_d\right)} \sum_{k=1}^{N_d} \frac{1}{1+\gamma d^2\left(k\right)} \tag{6}$$

where N_i is the ideal edge map's pixel edge numbers, N_d is the detected edge pixel numbers, $d\left(k\right)$ is the distance between the k-th detected edge pixel and the ideal pixel, is the constant (γ=1/4). The value of FOM ranges from 0 to 1, and being equal to 1 indicates a perfect match.

To test the proposed algorithm under noise, three levels of zero-mean Gaussian white noise are considered on test image 'Cameraman'. It is known that as the level of noise intensity increases, the values of the evaluation metrics become smaller, and the quality of the detected edge map becomes worse. Fig. 6 is the FOM evaluation values, the ideal edge map comes from the noise-free Canny result.

Besides the above images, we also use the RUG dataset. The BSDS500 [27] with 5-10 manually labeled ground-truths are mainly prepared for image segmentation methods, and therefore have been frequently used to evaluate learning-based edge detectors (due to the close relationship between them). The RUG dataset has 40 grayscale images with human-annotated ground-truth edges, which is more appropriate for testing traditional edge detectors, is used in our paper. The average P, R, $F_{0.5}$ of each algorithm are documented in Table I. For a subjective evaluation, edge maps of several RUG images yielded by various algorithms are shown in Fig. 7.

TABLE I
OBJECTIVE COMPARISON USING THE RUG DATASET

Algorithm	P	R	$F_{0.5}$
Canny [7]	0.1443	0.9905	0.2448
ED [24]	0.2854	0.9376	0.4118
CannyPF [25]	0.3438	0.8888	0.4677
RNFA [26]	0.4958	0.7939	0.5928
Proposed	**0.6491**	0.6395	**0.6138**

It is right that the proposed detector produced the lowest recall, and this can also be verified with the results presented in Fig. 7 in the paper. That is, our detector tends to suppress more insignificant edges so that a high precision can be obtained. In fact, there exists a trade-off between the precision and the recall for every edge detector shown in Table II, and we treat the F-measure as the most important metric in the evaluation. From Table I, the new detector yielded the highest F-measure.

TABLE II
THE TRADE-OFF BETWEEN THE PRECISION AND THE RECALL

Canny threshold	**Scale parameter**	P	R	$F_{0.5}$
default θ	$\sigma_1 = 1$, $\sigma_2 = 4$	0.2493	0.9592	0.3769
$\theta = 0.25$	$\sigma_1 = 1$, $\sigma_2 = 8$	0.6491	0.6395	0.6138
$\theta = 0.30$	$\sigma_1 = 1$, $\sigma_2 = 8$	0.7331	0.5549	0.6082

According to Table I and Fig. 7, it is obvious that our edge maps can keep the main edges, and are not sensitive to weak edges. On the other hand, other algorithms detect both main and redundant week edges. In the objective evaluation, our proposed algorithm achieves the highest precision value and a comparable recall value. These results confirm that the proposed algorithm can obtain more accurate edges, i.e. the detected edges are at their right locations, and derive comprehensive information of the input image, which are more in line with human visual characteristics.

V. CONCLUSION

In this work we have proposed a multi-scale edge detection algorithm based on a bi-directional edge tracing strategy. First, edge maps are generated at several given scales by using edge detector (e.g. Canny). Then the bi-directional tracing strategy is exploited which conducts a forward tracing operation to improve the accuracy of detected edges and a backward confirm operation to remove spurious edges. Experimental results show that our new algorithm outperforms a number of state-of-the-art edge detection algorithms in both subjective visual quality assessment and objective measurement.

REFERENCES

[1] W. Fu, M. Johnston and M. Zhang, "Low-level feature extraction for edge detection using genetic programming," *IEEE Transactions on Cybernetics*, vol. 44, pp. 1459–1472, 2014.
[2] Y. Liu, M. M. Cheng, X. Hu, K. Wang and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5872–5881.
[3] M. Seyedhosseini and T. Tasdizen, "Semantic image segmentation with contextual hierarchical models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 951–964, 2016.
[4] L. Sobel, "Camera models and machine perception," *Computer Science Department, Technion*, 1972.
[5] J. Prewitt, "Object enhancement and extraction," *Picture Processing and Psychopictorics*, vol. 10, pp. 15–19, 1970.
[6] D. C. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London*, vol. 207, pp. 187–217, 1980.
[7] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, 1986.
[8] C. Lopez-Molina, B. D. Baets, H. Bustince, J. Sanz and E. Barrenechea, "Multiscale edge detection based on Gaussian smoothing and edge tracking," *Knowledge-Based Systems*, vol. 44, pp. 101–111, 2013.

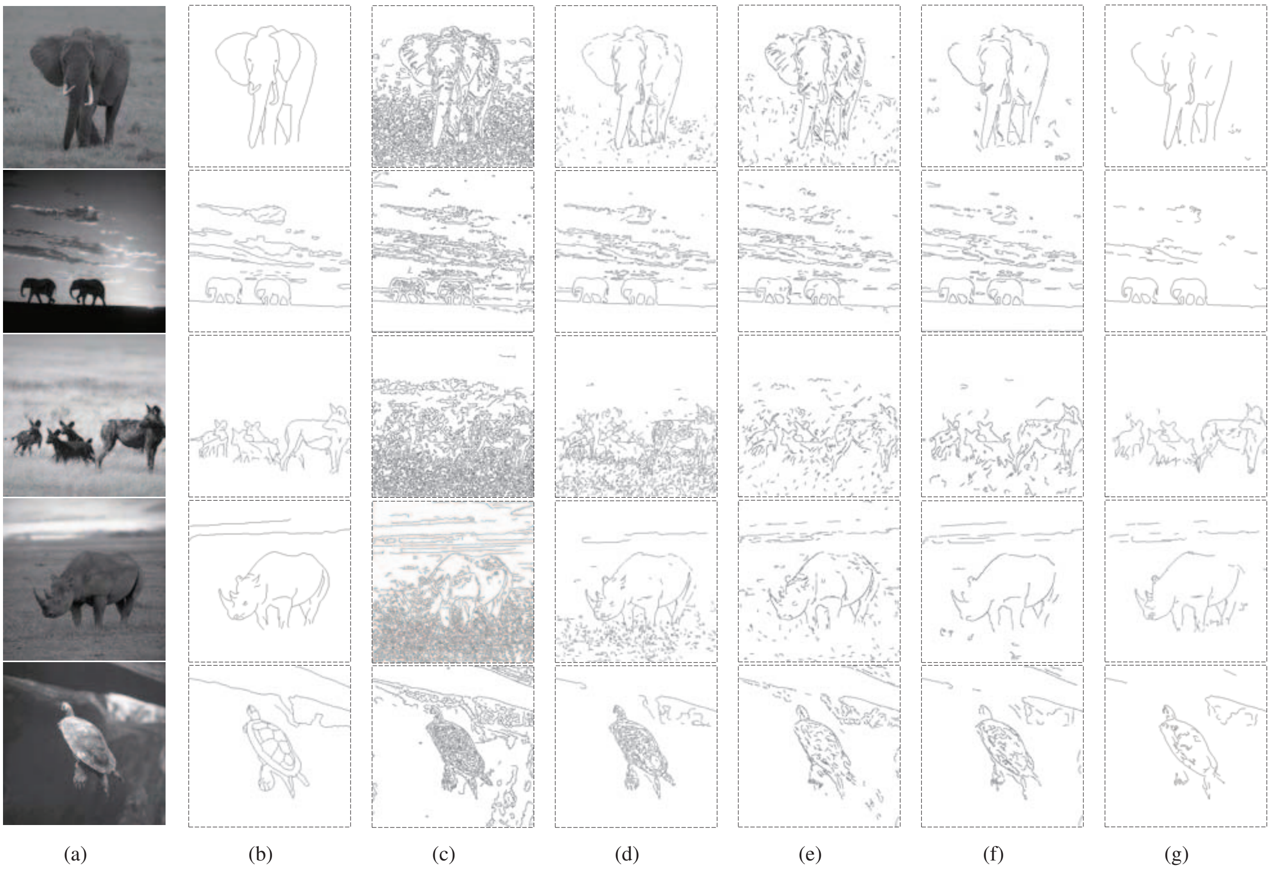


Fig. 7. Comparison of various edge detection algorithms on the RUG dataset. (a) Original image; (b) GT; (c) The Canny [7]; (d) The ED [24]; (e) The CannyPF [25]; (f) The RNFA [26]; (g) The proposed algorithm.

[9] S. Huang, J. Sun, Y. Yang, Y. Fang, P. Lin and Y. Que, "Robust Single-Image Super-Resolution Based on Adaptive Edge-Preserving Smoothing Regularization," *IEEE Transactions on Image Processing*, vol. 27, pp. 2650–2663, 2018.

[10] M. D. Ansari, A. R. Mishra and F. T. Ansari, "New divergence and entropy measures for intuitionistic fuzzy sets on edge detection," *International Journal of Fuzzy Systems*, vol. 20, pp. 474–484, 2018.

[11] Z. Yu, C. Feng, M. Y. Liu and S. Ramalingam, "CASENet: Deep category-aware semantic edge detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 21–26.

[12] Y. Liu and M. S. Lew, "Learning relaxed deep supervision for better edge detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 231–240.

[13] A. Karaali and C. R. Jung, "Edge-based defocus blur estimation with adaptive scale selection," *IEEE Transactions on Image Processing*, vol. 27, pp. 1126–1137, 2018.

[14] O. P. Verma and A. S. Parihar, "An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 25, pp. 114–127, 2017.

[15] P. Bao, L. Zhang and X. Wu, "Canny edge detection enhancement by Scale Multiplication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, p. 1485, 2005.

[16] S. Hallman and C. C. Fowlkes, "Oriented edge forests for boundary detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1732–1740.

[17] B. Gardiner, S. A. Coleman and B. W. Scotney, "Multiscale edge detection using a finite element framework for hexagonal Pixel-based images," *IEEE Transactions on Image Processing*, vol. 25, pp. 1849–1861, 2016.

[18] G. Wang, Z. Wang and J. Liu, "A new image denoising method based on adaptive multiscale morphological edge detection," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–11, 2017.

[19] A. P. Witkin, "Scale-space filtering," in *Proceedings of International Joint Conference Artificial Intelligence Karlsruhe*, 1983, pp. 1019–1022.

[20] P. Dollr and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[21] H. Jeong and C. I. Kim, "Adaptive determination of filter scales for edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 579–585, 1992.

[22] K. A. Ehinger, E. W. Graf, W. J. Adams and J. H. Elder, "Local depth edge detection in humans and deep neural networks," *2017 IEEE International Conference on Computer Vision Workshop*, 2017, pp. 2681–2689.

[23] B. Sumengen and B.S.Manjunath, "Multi-scale edge detection and image segmentation," in *Proceedings of 13th European Signal Processing Conference*, 2005, pp. 1–4.

[24] C. Topal and C.Akinlar, "Edge drawing: a combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, vol. 23, pp. 862–872, 2012.

[25] X. Lu, J. Yao, K. Li and L. Li, "Cannylines: A parameter-free line segment detector," in *Proceedings of IEEE International Conference on Image Processing*, 2015, pp. 507–511.

[26] X. Lu, J. Yao, L. Li, Y. Liu and W. Zhang, "Edge chain detection by applying Helmholtz principle on gradient magnitude map," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2017, pp. 1364–1369.

[27] D. R. Martin, C. C. Fowlkes and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 530–549, 2004.

[28] P. Dollr and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, pp. 1558-1570, 2015.